Article

Demosaicing by Differentiable Deep Restoration

Jie Tang ¹ , Jian Li ^{1,*} and Ping Tan ²
¹ College of Intelligence Science, National University of Defense Technology, Changsha 410073, China; kakaxi314@nudt.edu.cn

² School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada; pingtan@sfu.ca

* Correspondence: lijian@nudt.edu.cn

Abstract: A mosaic of color filter arrays (CFAs) is commonly used in digital cameras as a spectrally selective filter to capture color images. The captured raw image is then processed by a demosaicing algorithm to recover the full-color image. In this paper, we formulate demosaicing as a restoration problem and solve it by minimizing the difference between the input raw image and the sampled full-color result. This under-constrained minimization is then solved with a novel convolutional neural network that estimates a linear subspace for the result at local image patches. In this way, the result in an image patch is determined by a few combination coefficients of the subspace bases, which makes the minimization problem tractable. This approach further allows joint learning of the CFA and demosaicing network. We demonstrate the superior performance of the proposed method by comparing it with state-of-the-art methods in both settings of noise-free and noisy data.

Keywords: color filter array; demosaicing; color restoration; convolutional neural networks; closed-form optimization

Citation: Tang, J.; Li, J.; Tan, P. Demosaicing by Differentiable Deep Restoration. *Appl. Sci.* **2021**, *11*, 1649. <https://doi.org/10.3390/app11041649>

Academic Editor: Byung-Gyu Kim

Received: 4 January 2021

Accepted: 8 February 2021

Published: 12 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In digital imaging pipeline, a mosaic of color filter arrays (CFAs) is applied in front of the camera sensor to capture the raw image where each pixel only measures the intensity of one color band. This raw image measurement is referred to as imaging process and typically processed by a demosaicing algorithm to reconstruct a full-color image. The demosaicing process is a challenging ill-posed inverse problem with at least two-thirds of information missing in the raw image.

The demosaicing problem can be formulated as an image restoration problem, which aims to recover a full-color image I from the degraded raw image $M = HI + N$, where H is the degradation matrix and N is the additive noise. In contrast to other image restoration tasks, e.g., deblurring and denoising, where the matrices H are uncontrollable, the matrix H for demosaicing depends on the adopted CFA that is designable. Thus, both the design of CFA and demosaicing algorithm are critical for the quality of the final color image and have been studied for decades.

A variety of CFA patterns have been proposed and most demosaicing methods are developed accordingly to deal with raw images generated by the most popular CFA, e.g., Bayer pattern CFA [1]. As a limitation, these methods are difficult to generalize to images filtered by other CFAs. Recently, with a great success in many computer vision tasks, convolutional neural networks (CNNs) have been employed to solve the demosaicing problem by directly regressing the color image. This kind of learning-based methods presents impressive performance improvements and flexibility to fit any given CFA. However, since the tight coupling of imaging and demosaicing processes, learning a demosaicing network with a predefined CFA tends to be suboptimal. End-to-end learning of deep neural networks makes it possible to jointly learn the CFA and demosaicing algorithm. The pioneering works [2,3] proposed joint optimization by characterizing the CFA pattern as learnable parameters and simultaneously learn the CFA and demosaicing network.

However, in the inference stage, the demosaicing network only takes the raw image as input to regress the final color image, which does not make full use of the available CFA information.

Instead of applying CNNs to directly regress the color image, we design a deep neural network to solve the image restoration problem to find a color image I satisfying $M = HI + N$ according to the degradation model. The network learns to constrain the original ill-posed problem for robust solution, while the optimization module enforces the physical degradation model, i.e., the recovered color image I can produce the raw image M under the given CFA. In this way, the demosaicing network not only takes the raw image as input, but takes the degradation model with the adopted CFA into account as well. As proved by the studies in other vision tasks [4–6], this approach of combining machine learning and physically-based optimization often leads to superior results and better data generalization. At the same time, our whole network is differentiable, which also enables joint learning of the CFA and demosaicing network.

To this end, we introduce a differentiable deep restoration network for demosaicing. Specifically, we employ a network to predict the subspace where the color image I should lie in, i.e., generating multiple basis vectors and solving the color image I as a linear combination of these bases. In this way, the original ill-posed restoration problem $M = HI + N$ becomes well-posed and can be solved easily in closed-form. This closed-form solver minimizes the re-imaging error, i.e., $|M - HI|^2$.

Unlike LSM [4] and BA-Net [7] where the network produces basis vectors for the whole image and solves the output at once, we operate at image patch level to make the network training and generalization easier. Ideally, we can generate the bases for an $n \times n$ image patch, and solve the result image I within that patch. The whole image can then be solved by processing these overlapping $n \times n$ patches one by one and each patch determines the result of its center pixel. Interestingly, similar strategy is adopted in image matting [8] where the final alpha matte is assumed to be a linear combination of the R, G, B channels and a constant channel within local image patches. In comparison, our method further generalizes this formulation to employ a network to generate these bases. For better efficiency, our network does not generate bases for each $n \times n$ image patch separately. Instead, it produces some full-resolution bases and selects local patches from these full-resolution bases to solve I pixel by pixel.

We validate the proposed method on the challenging MIT dataset [9] and the well-known Kodak [10] and McMaster [11] datasets. Our method outperforms the state-of-the-art demosaicing methods in both settings of noise-free and noisy data. In addition to the learned CFA, it also works well with a fixed CFA, e.g., Bayer pattern CFA. Extensive ablation studies demonstrate the effectiveness of the proposed components and settings in our method. Moreover, our method presents a very fast running time, which is crucial for real-time applications. In summary, our contribution is two-fold: First, we present a novel method to solve demosaicing by differentiable deep restoration. The proposed method combines machine learning and physically-based optimization and shows superior performance compared to state-of-the-art solutions. Second, we introduce patch-based closed-form optimization scheme and jointly learn CFA and demosaicing network in end-to-end manner towards the optimal solution. Our code and trained models will be released at <https://github.com/kakaxi314/D3R> (accessed on 12 February 2021).

2. Related Work

Both the CFA and demosaicing algorithm have large impact on the quality of the final result. In this section, we review previous methods on the design of CFA and demosaicing.

2.1. CFA Design

The Bayer pattern [1] is the most commonly used CFA which is designed according to the spectral sensitivity of the human visual system. Several other CFAs have been proposed empirically for various considerations. Luminance channel was introduced to reduce the

exposure time with high sensitivity in [12]. CFA with randomly sampled colors [13–15] was proposed to reduce the aliasing artifacts. In addition to the periodic tiling CFA pattern, Bai et al. [16] developed an irregular CFA with aperiodic tiling Penrose layout. Rather than designing CFA pattern empirically, some CFAs were optimized by minimizing the reconstruction error for a given demosaicing algorithm. A perceptual error was proposed in [17] as optimization objective to select color from R, G, B channels for CFA pattern. In [18], the CFA was iteratively optimized for the proposed equivalent conditions.

In frequency domain [19], the raw image can be seen as multiplexing of one luminance component at baseband and several chrominance components at high frequency bands. A variety of methods have been proposed to reduce aliasing caused by CFA by maximizing distance among these components in frequency domain. The design of CFA was formulated as a constrained optimization problem. Hirakawa et al. [20] optimized the CFA by parameter selection in the frequency domain with explicitly designed constraints. Hao et al. [21] utilized a geometric method to solve the constrained optimization problem of the specified frequency structure. Bai et al. [22] proposed an automatic CFA generation method with no requirement of human interaction.

Although the above CFA design methods produce good image results, they are either designed without considering the demosaicing algorithm or designed for only a specific kind of demosaicing algorithm. In comparison, we jointly learn the CFA and demosaicing network for best performance.

2.2. Demosaicing

Most of the traditional demosaicing methods are developed to deal with the raw image filtered by a certain CFA, e.g., Bayer pattern [1]. They can be roughly split into spatial domain methods and frequency domain methods.

Among spatial domain methods, bilinear interpolation from the nearby pixels of the same color channel is the simplest one, which works well in flat image regions, but suffers from severe artifacts and losses in details. Some linear demosaicing methods [23,24] working in a data-driven way by Linear Minimum Mean Square-Error Estimation (LMMSE) achieves good performance with fast running time. Rather than processing each color channel independently, some interpolation methods [25,26] focused on inter-channel correlation based on the constant color difference assumption. The edge information was utilized in demosaicing techniques [27–29] to alleviate the artifacts caused by interpolation across edges. Some other properties like self-similarity [30] and data-redundancy [31] were exploited to improve the quality of reconstruction. Post-processing method [32] was also studied to improve the image quality of the initially demosaiced image.

Among frequency domain methods, several techniques were developed to estimate the components in frequency domain to recover color image. In [33], the demosaicing method was designed for raw image generated by Bayer pattern, in which green color owns the most sample rate. Green component was estimated first by a diamond shape 2D filter in frequency domain, then the high-frequency information of green channel was added to improve the reconstruction of red and blue channels. Some demosaicing methods were proposed for more general situation, without restriction to one certain CFA. Luminance and chrominance components were directly estimated by appropriate frequency selection in [19], and the color image was recovered from these frequency components. We refer [34] to have a complete review of traditional demosaicing methods.

Recently, deep neural networks present impressive improvements on the demosaicing task. Gharbi et al. [9] proposed a joint approach for demosaicing and denoising by using CNN to directly regress the color image. Tan et al. [35] designed a two-stage demosaicing architecture for Bayer pattern CFA. The green channel was recovered in the first stage and the red and blue channels were estimated by residual learning in the second stage. Based on the observation that the correlations between different channels are quite different, Cui et al. [36] proposed a 3-stage CNN structure for demosaicing, in which two separated networks were used to reconstruct red and blue channels. Huang et al. [37]

designed an effective feature extraction network for joint demosaicing and denoising. Liu et al. [38] exploited green channel and density map as guidance in network structure for joint demosaicing and denoising. In contrast to most traditional demosaicing methods, demosaicing methods with deep neural networks are more flexible with the ability to work on raw image filtered by any CFA.

To the best of our knowledge, the methods of Kokkinos et al. [39,40] is the only deep learning based demosaicing method that takes the degradation model into account. They trained an iterative network for joint demosaicing and denoising based on the given CFA. Thus, the demosaicing method was still optimized isolated from the design of CFA. In contrast, we go one step further, the CFA pattern is jointly learned in our method, which owns more benefits. Furthermore, our method is more convenient to train and more effective and efficient to execute by directly generating the color image in a closed-form solution other than in an iterative way.

2.3. Joint Optimization of CFA and Demosaicing

Recently, with the help of end-to-end learning of deep neural networks, pioneering works with joint optimization of CFA and demosaicing have been proposed, where the CFA is learnable parameters jointly optimized with the demosaicing network. Chakrabarti et al. [2] proposed the first joint optimization framework, in which the design of CFA was modeled as a color channel selection at each pixel from the predefined color set. In [3], the jointly learned CFA pattern was regressed directly which is able to run throughout the whole color space. While CFA is jointly optimized with demosaicing network in these two methods, the demosaicing network only takes the raw data as input to directly regress the color image. In contrast, our method is a combination of machine learning and physically based optimization, that the demosaicing network not only takes the raw image as input, but takes the degradation model and the adopted CFA into account as well. The prediction is constrained to satisfy the degradation model by a closed-form solver, which is differentiable that our method can also follow the joint optimization paradigm.

3. The Proposed Method

3.1. Overview

We design a deep restoration method for demosaicing that estimates the color image I by minimizing

$$\|M - HI\|^2. \quad (1)$$

This problem is under-constrained even H is known, due to the sub-sampling nature of the degradation model.

The pipeline of the proposed method is illustrated in Figure 1. The input raw image goes through a Pattern Sharing Convolution (PSC) layer as a preprocessing. This preprocessing is necessary, because the data in the raw image is sparse where each pixel has only the intensity of one color band. This is not appropriate for CNNs to directly operate on. After that, we use a U-Net to extract image features and generate some bases $B_1, B_2, \dots, B_m$. We consider the result I to be a linear combination of these bases within local image patches, i.e., $I^{(i)} = v_1^{(i)} B_1^{(i)} + v_2^{(i)} B_2^{(i)} \dots + v_m^{(i)} B_m^{(i)}$. Here, the superscript (i) indicates the i -th $n \times n$ local image patch ($n = 5$ in our experiments). Here, both the full-color image patch $I^{(i)}$ and the bases patches $B_j^{(i)}$ are 3-channel image patches. We solve the combination coefficients $V^{(i)} = [v_1^{(i)}, v_2^{(i)}, \dots, v_m^{(i)}]^\top$ by minimizing Equation (1) in closed-form solution. Note that the coefficient $v_1^{(i)}, v_2^{(i)}, \dots, v_m^{(i)}$ are scalars, which means they are shared by the red, green, and blue channels. We choose this design for strong correlation between different color channels.

Figure 1. The overall architecture of our deep restoration network. The whole architecture consists of imaging process and demosaicing process. For imaging process, the input image is filtered by the color filter arrays (CFA) to produce the raw image. For demosaicing process, the prediction is reconstructed with basis maps generated from the U-net structure and the corresponding coefficient maps optimized by a differentiable closed-form solver. The whole architecture is trained end-to-end.

As illustrated in Figure 2, we slide these patches over the image and solve the result at the patch center. In our implementation, the processing of different patches is parallelized by GPU processing.

Figure 2. Example of a color image reconstruction from the estimated basis maps and the corresponding optimized coefficient maps. The value of basis maps and the value of coefficient maps have been normalized for visualization.

3.2. Model the Imaging Process

We can model the imaging process as a linear process depending on the color image and the corresponding CFA along the color channels. Most of the existing CFAs are periodic tiling of square pixels with the same size of the image resolution. For a CFA pattern P with size of $k \times k$, the CFA is:

$$H_c(i) = P_c(i \bmod k), \text{ for } c \in C, \quad (2)$$

where i is the 2D pixel coordinates with x and y coordinates, C is the set of color channels $\{R, G, B\}$ and mod indicates the modulus operator which works element-wisely. To guarantee the designed CFA is physically realizable, P_c takes values between $[0, 1]$ and satisfies

$$P_R + P_G + P_B = \mathbf{1}. \quad (3)$$

Here, the $\mathbf{1}$ is an all-one matrix, where all matrix elements equals 1. When the CFA is given, the mosaiced image M obtained from this imaging process can be formulated as

$$M(i) = \sum_c H_c(i) \cdot I_c(i) + N(i). \quad (4)$$

Here, N is the additive noise. For noise-free data in ideal conditions, N can be ignored in this imaging process. Otherwise, we need to model the additive noise N to simulate the raw image from color image. The potential influence of float precision of H and N is not considered in the simulation process.

In this imaging process, the choice of the CFA pattern directly affects the quality of the raw image M we obtained from the sensor. Plenty of CFAs have been proposed in the long time study. The Bayer pattern [1] Ba is the most popular CFA pattern with 2×2 local window, where the color is a selection of primary colors of R, G, B for each pixel. According to our formulation, this CFA pattern can be written as,

$$P_R^{Ba} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, P_G^{Ba} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, P_B^{Ba} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \quad (5)$$

In addition to the primary colors, the color for CFA pattern can also be chosen from other spectral bands. For example, the Bean pattern [41] Be is designed as,

$$P_R^{Be} = \begin{bmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}, P_G^{Be} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, P_B^{Be} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{bmatrix}. \quad (6)$$

According to Equation (3), the sum of the CFA H across all channels should be an all-one matrix. Furthermore, all elements in a CFA must be non-negative. These two constraints can be written as

$$\begin{cases} \sum_c H_c = \mathbf{1}, \\ H \geq \mathbf{0}. \end{cases} \quad (7)$$

In the proposed method, in which the CFA pattern is learnable and can be optimized jointly with the demosaicing network, a Softmax layer with positive and normalized outputs is employed on the learned parameters to guarantee the generated CFA satisfies the above requirements. As pointed by [3], this kind of CFA which is a linear combination of primary colors can be manufactured by technology described in [42]. One of the CFA pattern learned by the proposed method is shown in Figure 1.

3.3. Differentiable Deep Restoration

Previous deep learning based methods for demosaicing often address it by direct regression. In comparison, we represent the result color image I as a linear combination of basis maps within a local window centered at every pixel, which can be expressed as

$$I^{(i)} = B^{(i)\top} V^{(i)}. \quad (8)$$

Here, $^{(i)}$ standards for the local window centered at the i -th pixel. The basis maps $B^{(i)}$ are produced by a neural network, while the combination coefficients $V^{(i)}$ are online optimized by minimizing the cost function in Equation (1). Thus, the demosaicing process works in a hybrid architecture. The network generates basis maps according to image context to regularize the solution, and the optimization solves the inverse problem to enforce the physically-based degradation model.

Therefore, the coefficient for a pixel i is

$$V^{(i)} = \arg \min_X \sum_{j \in \mathbb{N}(i)} \|H(j)B^{(i)\top}(j)X - M(j)\|^2 + \lambda \|X\|^2, \quad (9)$$

where $\mathbb{N}(i)$ is the set of pixels in the $n \times n$ local window centered at i , and the coefficient λ is the factor of the regularization term. This minimization problem is efficiently solved by the Cholesky decomposition with closed-form solution. This closed-form solver is differentiable and can be embedded into the network without any difficulty. The regularization term here is to avoid instable results caused by perturbations in bases. It also prevents the numerical instability of matrix inverse in the optimization. In order to find appropriate settings of n and λ , we fix basis number m as 4, and change n from $\{3, 5, 7\}$ and λ from $\{0.001, 0.01, 0.1\}$. Finally, we set $n = 5$ and $\lambda = 0.01$ empirically for our method.

An example of some learned bases and corresponding coefficients is visualized in Figure 2. The bases for each local window is cropped from the full resolution basis maps generated by our basis generation network. The whole image can then be solved by processing these overlapping patches one by one and each patch determines the result of its center pixel.

3.3.1. Bases Generation Network with Sparse Data

One challenge in designing the bases generation network is to deal with the raw image M with missing entries. For example, as shown in Figure 3, in the raw image generated by the Bayer pattern, only one of the four neighboring pixels have red intensity measured. The red intensity is missing at all the other three pixels. This sparsity makes the raw image M inappropriate for conventional CNNs with weight sharing for different pixels.

Figure 3. Example of CFA patterns and the corresponding raw images. The top row shows the Bayer pattern, the second row is our learned CFA pattern. The values of a CFA are marked in the CFA cells.

Various preprocessing methods have been proposed to deal with this sparsity. The most commonly used preprocessing operations are rearranging and interpolation. The

rearranging operation is adopted in [9] to convert the raw image sampled from a 2×2 Bayer pattern as the following,

$$D_c(i) = M(k \times i_x + c \bmod k, k \times i_y + \lfloor \frac{c}{k} \rfloor). \quad (10)$$

Basically, it reduces the image resolution and packs all pixels with red, green, or blue intensities to form the R, G, B channels, respectively. This operation is computational efficient and also reduces the computation of the subsequent networks, since it reduces the image resolution. However, when the size of the CFA pattern enlarges, which tends to improve the image quality as reported in [22], this rearranging operation loses spatial information due to the shrinkage of image resolution. In comparison, the interpolation operation is used in paper [35] to preprocess the raw image with hand-crafted interpolation weights. The missing R, G, B values are linearly interpolated from the corresponding channel in a neighborhood, which can be written as

$$D_c(i) = \sum_o W_c(o) M(i + o) \text{Mask}_c(i + o), \quad (11)$$

where W is the hand-crafted interpolation weights, o is a coordinate offset within the local window, and Mask_c is a binary mask indicating whether the pixel has values on channel c . However, hand-crafted interpolation weights could be sub-optimal and degrades the demosaicing results. Interpolating across edges may introduce large errors and make the subsequent processing difficult.

In comparison, we propose a novel pattern sharing convolutional (PSC) layer to allow the network to deal with the sparse image M . Mathematically, our processed data is

$$D_c(i) = \sum_o W_c^l(i \bmod k, o) M(i + o), \quad (12)$$

where the W^l are parameters learned with the whole network. As the color channels of raw image are periodic, the parameters are shared with the same period k . Basically, this PSC layer learns interpolation parameters instead of the hand-crafted ones in Equation (11). Furthermore, it takes intensity values from all color channels in the neighborhood to interpolate the missing entries.

Compared with the rearranging operation, our PSC layer can take the advantage of the neighborhood relationship information in local and maintain the resolution of image which makes it suitable for CFA pattern with large size. Compared with the hand-crafted interpolation operation, our PSC layer is a more general way to process the sparse data whose parameters can be learned with the whole network in an end-to-end manner. More comparisons and discussions can be found in detail in Section 4.2.

3.3.2. Bases Generation Network with U-Net Structure

Most deep learning based demosaicing methods use a CNN structure extracting features at the original input resolution to directly regress the color image. In contrast, we adopt a U-net structure to generate our basis maps from the preprocessed data D as,

$$B = f(D; \Theta). \quad (13)$$

Here, f indicates the non-linear convolutional neural network and Θ are the network parameters. By extracting features on multi-resolution, the U-net structure can easily achieve large receptive field without consuming too much computation and GPU memory.

We adopt a U-Net [43] of fully convolutional network [44] with four different resolutions to extract features and generate the full-resolution basis maps, as illustrated in Figure 1. In this U-Net structure, we employ additional skip connections to fuse the features with the same resolution from the encoder to the decoder. At each resolution, the encoder consists of 2 stacked ResBlocks. A ResBlock is a residual block with two sequential 3×3 convolution

layers from [45]. Each convolution layer has replicated padding ahead it and is followed by a batch normalization layer [46].

3.4. Joint Learning CFA and Demosaicing Network

Our deep restoration network is differentiable and can be trained in an end-to-end manner. This brings two advantages. First, the closed-form solver enforces the estimated color image I to satisfy the physical degradation model. Second, as the CFA H is explicitly included in the restoration formulation, the back-propagated gradient can quickly reach the CFA without having to go through the entire demosaicing network. In this way, our method can provide better supervision to train the CFA. In comparison, the pioneer methods [2,3] in jointly optimizing CFA and demosaicing networks could suffer from the notorious gradient vanishing problem as the back-propagation has to pass through the demosaicing network before reaching the CFA.

3.5. Training Settings

We adopt the mean squared error (MSE) as the training loss to train the network on the whole training set of MIT dataset [9] with 2,590,185 color images of 128×128 resolution. Random rotation of 90 degrees and random flips along horizontal and vertical axes are used as data augmentation. We utilize ADAM [47] as the optimizer with an original learning rate of 10^{-3} and weight decay of 10^{-2} . The learning rate drops to half every 3 epochs. We train the network from scratch with batch size of 32 by one Nvidia GTX 1080Ti GPU. The code is implemented in Pytorch.

4. Experiments

We conduct comprehensive experiments on benchmark images to evaluate and analyze our method. We first compare our method with other state-of-the-art demosaicing technologies on noise-free data. Then, extensive ablation studies are designed to investigate the impact of the proposed components and related settings in our method. As the raw data captured by sensor is corrupted by noise in practice, we evaluate our method with noisy data in the next. Finally, the running time which is crucial for real-time applications is also tested and compared. The main metric used for evaluation is the average peak signal-to-noise ratio (PSNR), where the MSE is calculated over pixels and color channels before taking logarithmic scale. We also report average structural similarity (SSIM) [48] which measures the structure fidelity. The average SSIM is calculated by averaging the SSIM values individually from the R, G, and B channels. For both PSNR and SSIM, the higher value indicates better perceptual quality.

4.1. Reconstruction from Noise-Free Data

We first compare the proposed method with state-of-the-art methods on noise-free data. Our method is trained on the training set of the MIT dataset [9] consisting of the *vdv* and *moiré* datasets. We train our network with a fixed Bayer pattern and a jointly learned CFA separately for evaluation. We evaluate different demosaicing techniques on the test set of the *vdv* and *moiré* datasets as well as the commonly used Kodak [10] and McMaster [11] datasets. For our network, as the resolution of input image is required to be a multiply of the downsampling factor of U-Net, we employ replicated padding on the input image to satisfy this requirement and crop the prediction to the original resolution in the end for evaluation.

Table 1 lists the average PSNR and average SSIM of different methods on these four datasets. Techniques based on Bayer pattern CFA are compared in the top part, while in the bottom are techniques with non-Bayer CFAs. For fair comparison, we evaluate these methods with the source code and the trained models provided by authors if they are available. All result images are saved to disk to have the same quantization (i.e., 0–255) and then calculated PSNR and SSIM. Otherwise, we directly quote the evaluation result

from the published paper [20–22,49]. For methods in [35,50], whose reconstruction is not the full image, we only evaluate within the reconstructed area.

Our method produces the best PSNR in all the four datasets with both settings of Bayer pattern and non-Bayer CFA. Our strong performance in these four datasets demonstrates the advantages of our design of combining machine learning and physically-based optimization. Meanwhile, our method with jointly learned CFA performs much better than the setting of Bayer pattern. This verifies the importance of CFA design and the advantages of jointly optimizing the CFA pattern and the demosaicing algorithm. While our method is only trained on the training set of the MIT dataset [9] where the image resolution is fixed at 128×128 , it has strong generalization capability to deal with the Kodak [10] and McMaster [11] datasets with high resolution images. Instead of only measuring pixel-intensity differences as PSNR, SSIM considers the fidelity of image structure. We can see from Table 1 that our method also achieves comparable and even higher SSIM than other demosaicing techniques.

Table 1. PSNR (peak signal-to-noise ratio) (dB) and SSIM (structural similarity) results on noise-free data. The comparison methods are evaluated on the Kodak, McMaster, vdp, moiré datasets.

Demosaic (Bayer CFA)	Kodak (24 images)		McMaster (18 images)		vdp (1000 images)		moiré (1000 images)	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bilinear	29.26	0.881	30.81	0.922	23.77	0.806	25.27	0.820
Condat et al. [51]	34.81	0.954	30.20	0.861	27.34	0.886	29.90	0.883
Tan et al. [35]	41.98	0.988	38.94	0.969	32.99	0.955	34.28	0.930
Gharbi et al. [9]	41.50	0.987	39.14	0.971	33.95	0.973	36.62	0.960
Cui et al. [36]	42.18	0.988	39.33	0.972	33.23	0.960	34.74	0.934
Huang et al. [37]	42.34	0.989	39.10	0.970	33.46	0.959	34.99	0.935
Henz et al. [3]	41.93	0.988	39.51	0.972	34.30	0.973	36.41	0.956
Kokkinos et al. [40]	41.65	0.989	39.51	0.971	34.46	0.966	36.93	0.956
Ni et al. [32]	40.36	0.986	38.11	0.967	31.54	0.943	32.95	0.926
Our 2×2 Bayer	42.49	0.989	39.76	0.972	35.04	0.977	37.54	0.952
(Non-Bayer CFA)	Kodak		McMaster		vdp		moiré	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Chakrabarti et al. [50]	33.51	0.962	30.95	0.921	25.91	0.897	28.77	0.910
Condat et al. [52]	39.96	0.986	33.88	0.931	30.04	0.934	33.33	0.927
Hao et al. [21]	39.42	—	—	—	—	—	—	—
Bai et al. [22]	40.24	—	—	—	—	—	—	—
Hirakawa et al. [20]	40.36	—	—	—	—	—	—	—
Li et al. [49]	41.59	—	—	—	—	—	—	—
(Joint-Learned CFA)	Kodak		McMaster		vdp		moiré	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Chakrabarti et al. [2]	31.35	0.920	28.39	0.835	24.98	0.824	27.91	0.845
Henz et al. [3]	43.13	0.991	40.18	0.976	35.17	0.977	37.70	0.961
Our Learned 4×4	43.87	0.992	40.60	0.976	36.14	0.981	39.28	0.967

In addition, we compare the result quality of our method with other demosaicing techniques visually in Figure 4. From left to right, the different columns show the ground truth reference image, the results by the methods of Condat et al. [52], Gharbi et al. [9], Kokkinos et al. [40], Henz et al. [3], and our method with jointly learned CFA pattern and demosaicing algorithm. It is clear that our method presents better results with more details, especially in the high frequency regions. While the other methods often suffer from color artifacts on these challenging examples.

Figure 4. Performance comparison on noise-free data. The first column shows the reference images. The predictions of our method are shown in the last column and compared with Condat et al. [52], Gharbi et al. [9], Kokkinos et al. [40] and Henz et al. [3]. The notable regions are selected and zoomed in with cyan rectangles. Best viewed in digital version.

4.2. Ablation Studies

To investigate the importance of the different components and various settings in our method, we perform ablation studies with the joint CFA and demosaicing optimization on noise-free data. The dataset for training is still the MIT dataset [9].

4.2.1. Sparse Data Processing and Optimization

We first investigate the effectiveness of different sparse data processing. For this experiment, we directly regress the color image and adopt the rearranging operation [9] and the bilinear interpolation [35] to preprocess the raw image, respectively. Since the rearranging operation reduces the image resolution, for this variant, all the following ResBlocks work on the same resolution without using the U-Net structure to further reduce the resolution of intermediate features. Second, we investigate the effectiveness of the optimization based restoration and compare it with direct regression. For this comparison, the closed-form solver in our method is replaced by directly regressing the color image.

We can see the PSNR results in Table 2. It is clear that our preprocessing with the PSC layer is more effective than the rearranging or interpolation, as the PSNR associated with the ‘Interpolation’ and ‘Rearranging’ variants are lower than the ‘PSC’ variant. Replacing the restoration based optimization by direct regression also causes a PSNR drop.

Table 2. Ablation study for variations on the test set of the MIT dataset [9].

Preprocessing	Optimization	Basis Number	PSNR
Rearranging	No	—	36.68
Interpolation	No	—	36.79
PSC	No	—	36.90
PSC	Global	4	37.61
PSC	Global	8	37.64
PSC	Global	12	37.67
PSC	Global	16	37.43
PSC	Local	4	37.71

4.2.2. Different Basis Number

For each pixel, our result is a linear combination of multiple bases predicted by the network. The number of bases is fixed and consistent for all the pixels. In this experiment, we analyze the effect of the number of these bases. We plot the PSNR curve on the test set of MIT dataset [9] with different number of basis maps m in Figure 5. We experiment with $m = 1, 4, 8, 12, 16$ to train our method separately. We can see that the PSNR increases with the enlarging of the number of bases.

However, a larger number of bases also means more computation. Thus, we also plot the number of FLOPs of the differentiable closed-form solver for different number of bases. The FLOPs are approximately estimated by the computation complexity of the solver, which is $\mathcal{O}(m^3)$ depending on the number of bases. We can see the computation increases dramatically when the number of bases increases, while the improvement on result quality gradually saturates. Thus, to take a trade-off between result quality and computation complexity, we fix the number of bases $m = 4$ in our method.

Figure 5. The performance of our method with different number of bases. The left axis is the PSNR on the test set of MIT dataset [9]. The right axis is the extra FLOPs caused by the differentiable closed-form solver.

4.2.3. Effect of Patch-Based Optimization

In our method, the combination coefficients of different bases are solved at overlapping local image patches. Instead of solving these coefficients patch by patch, an alternative way

is to solve the set of combination coefficients for the whole image as in the work LSM [4]. We refer this kind of optimization as ‘Global’. As shown in Table 2, our method with coefficients solved at local windows (which is referred as ‘Local’) generates higher PSNR than ‘Global’. We further vary the number of bases for the ‘Global’ method. While a larger number of bases sometimes improves the result quality and produces similar results to our local method, we also find it makes the training process less stable in experiments.

4.3. Reconstruction from Noisy Data

Real images are often contaminated by noises. In this experiment, we assume additive Gaussian noise [53] and experiment different methods under different noise levels. We use an image with additive Gaussian noise to generate the raw image M . Our goal is to reconstruct the original image I from the noise corrupted data.

We experimented 5 different levels of Gaussian noise. To verify the capability of our method on noisy data, without any modification on network architecture, we directly train our demosaicing network with jointly learned CFA from scratch on the noisy image. In the training stage, the input image is contaminated with a random level of Gaussian noise. Then the trained network is tested on noisy data contaminated by these 5 levels of Gaussian noise separately. Specifically, the standard deviations σ of the Gaussian noise is set to $\{4, 8, 12, 16, 20\}$.

Figure 6 compares the PSNR of our method with other techniques on the test set of MIT dataset [9] corrupted by different levels of Gaussian noise. We can see that our method presents good generalization capability for different levels of noise with a single model. It outperforms other joint denoising and demosaicing methods, even the denoising problem is not explicitly considered in our network design.

Figure 6. PSNR comparison on the test set of MIT dataset [9] corrupted by Gaussian noise with standard deviations σ of $\{4, 8, 12, 16, 20\}$. Our method is compared with joint demosaicing and denoising methods Chakrabarti et al. [2], Condat et al. [52], Gharbi et al. [9] and Henz et al. [3].

We also show the reconstructed color image from noisy data in Figure 7. The first and last columns are the corrupted image and the clean reference image, respectively. The other four columns are results from different methods. We can see that our method can successfully reduce the noise in the corrupted image. It presents clear color reconstruction without oversmoothing the high frequency content.

Figure 7. Performance on the noisy data under different noise levels. The first column shows the corrupted images and the last column shows the reference images. The predictions of our method are visually compared with Condut et al. [52], Gharbi et al. [9] and Henz et al. [3]. The notable regions are selected and zoomed in with cyan rectangles. Best viewed in digital version.

4.4. Running Time Comparison

We test the running time of our method and compare it with other state-of-the-art demosaicing techniques with public implementation. All these methods are tested with the most commonly used Bayer pattern CFA on a computer workstation with Nvidia GTX 1080Ti GPU and Intel(R) Xeon(R) CPU E5-2603. The running time is measured on an image with one million pixels. We use the average running time of 20 tests. As shown in Table 3, our method is highly efficient among these deep learning based methods. It presents comparable running time with other direct regression methods [3,9], and is much faster than the iterative method [40]. Furthermore, in contrast to some iterative methods [54] with stopping criterion, which makes the running time data dependent, our method has constant running time which is attractive in real applications.

Table 3. Running Time Comparison on an image with one million pixels.

	Time (ms/Mpix)
Gharbi et al. [9]	277
Henz et al. [3]	719
Kokkinos et al. [40]	1177
Ours	365

5. Conclusions

In this paper, we propose a novel end-to-end learned network that is a combination of machine learning and physically-based optimization to address the image demosaicing

problem. The demosaicing network contains a subspace learning component to constrain the solution and an optimization component to enforce the degradation model and solve the ill-posed restoration problem. Our method jointly optimizes the demosaicing network and the color filter array (CFA) for better performance. Extensive experiments demonstrate the superior performance of our method with fast running time. While this paper specifically focuses on the demosaicing issue of color image restoration, we hope the insights in our method can be extended to numerical restoration problems of image data (e.g., biology image, satellite image, etc), and other image processing problems (e.g., image denoising, deblurring, etc).

Author Contributions: Conceptualization, J.T. and J.L.; methodology, J.T. and P.T.; software, J.T.; validation, J.T.; writing—original draft preparation, J.T. and J.L.; writing—review and editing, J.T. and P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under Grant 61973311.

Data Availability Statement: Code and trained models are available at <https://github.com/kakaxi314/D3R>.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bayer, B.E. Color Imaging Array. U.S. Patent 3,971,065, 20 July 1976.
2. Chakrabarti, A. Learning sensor multiplexing design through back-propagation. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3081–3089.
3. Henz, B.; Gastal, E.S.; Oliveira, M.M. Deep joint design of color filter arrays and demosaicing. *Comput. Graph. Forum* **2018**, *37*, 389–399. [\[CrossRef\]](#)
4. Tang, C.; Yuan, L.; Tan, P. LSM: Learning Subspace Minimization for Low-level Vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, Seattle, WA, USA, 13–19 June 2020.
5. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3929–3938.
6. Li, L.; Pan, J.; Lai, W.S.; Gao, C.; Sang, N.; Yang, M.H. Learning a discriminative prior for blind image deblurring. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6616–6625.
7. Tang, C.; Tan, P. BA-Net: Dense Bundle Adjustment Networks. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
8. Levin, A.; Lischinski, D.; Weiss, Y. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *30*, 228–242. [\[CrossRef\]](#)
9. Gharbi, M.; Chaurasia, G.; Paris, S.; Durand, F. Deep joint demosaicking and denoising. *ACM Trans. Graph. (TOG)* **2016**, *35*, 191. [\[CrossRef\]](#)
10. Franzen, R. Kodak Lossless True Color Image Suite. 1999. Volume 4. Available online: <http://r0k.us/graphics/kodak> (accessed on 13 June 2020).
11. Zhang, L.; Wu, X.; Buades, A.; Li, X. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *J. Electron. Imaging* **2011**, *20*, 023016.
12. Yamagami, T.; Sasaki, T.; Suga, A. Image Signal Processing Apparatus Having a Color Filter with Offset Luminance Filter Elements. U.S. Patent 5,323,233, 21 June 1994.
13. Zhu, W.; Parker, K.; Kriss, M.A. Color filter arrays based on mutually exclusive blue noise patterns. *J. Vis. Commun. Image Represent.* **1999**, *10*, 245–267. [\[CrossRef\]](#)
14. Condat, L. A new random color filter array with good spectral properties. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 1613–1616.
15. Oh, P.; Lee, S.; Kang, M.G. Colorization-based RGB-white color interpolation using color filter array with randomly sampled pattern. *Sensors* **2017**, *17*, 1523. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Bai, C.; Li, J.; Lin, Z.; Yu, J.; Chen, Y.W. Penrose demosaicking. *IEEE Trans. Image Process.* **2015**, *24*, 1672–1684. [\[PubMed\]](#)
17. Parmar, M.; Reeves, S.J. A perceptually based design methodology for color filter arrays [image reconstruction]. In Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, QC, Canada, 17–21 May 2004; Volume 3, pp. iii–473.
18. Lu, Y.M.; Vetterli, M. Optimal color filter array design: Quantitative conditions and an efficient search procedure. In Proceedings of the Digital Photography V, International Society for Optics and Photonics, San Jose, CA, USA, 5 June 2009; Volume 7250, p. 725009.

19. Alleysson, D.; Susstrunk, S.; Hérault, J. Linear demosaicing inspired by the human visual system. *IEEE Trans. Image Process.* **2005**, *14*, 439–449.
20. Hirakawa, K.; Wolfe, P.J. Spatio-spectral color filter array design for optimal image recovery. *IEEE Trans. Image Process.* **2008**, *17*, 1876–1890. [[CrossRef](#)] [[PubMed](#)]
21. Hao, P.; Li, Y.; Lin, Z.; Dubois, E. A geometric method for optimal design of color filter arrays. *IEEE Trans. Image Process.* **2010**, *20*, 709–722.
22. Bai, C.; Li, J.; Lin, Z.; Yu, J. Automatic design of color filter arrays in the frequency domain. *IEEE Trans. Image Process.* **2016**, *25*, 1793–1807. [[CrossRef](#)]
23. Amba, P.; Dias, J.; Alleysson, D. Random color filter arrays are better than regular ones. In Proceedings of the Color and Imaging Conference, San Diego, CA, USA, 7–11 November 2016; Volume 2016, pp. 294–299.
24. Amba, P.; Thomas, J.B.; Alleysson, D. N-LMMSE demosaicing for spectral filter arrays. *J. Imaging Sci. Technol.* **2017**, *61*, 40407–1. [[CrossRef](#)]
25. Pei, S.C.; Tam, I.K. Effective color interpolation in CCD color filter arrays using signal correlation. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 503–513.
26. Li, X. Demosaicing by successive approximation. *IEEE Trans. Image Process.* **2005**, *14*, 370–379.
27. Li, X.; Orchard, M.T. New edge-directed interpolation. *IEEE Trans. Image Process.* **2001**, *10*, 1521–1527. [[PubMed](#)]
28. Kakarala, R.; Baharav, Z. Adaptive demosaicing with the principal vector method. *IEEE Trans. Consum. Electron.* **2002**, *48*, 932–937. [[CrossRef](#)]
29. Lu, W.; Tan, Y.P. Color filter array demosaicking: New method and performance measures. *IEEE Trans. Image Process.* **2003**, *12*, 1194–1210.
30. Buades, A.; Coll, B.; Morel, J.M.; Sbert, C. Self-similarity driven color demosaicking. *IEEE Trans. Image Process.* **2009**, *18*, 1192–1202. [[CrossRef](#)]
31. Mairal, J.; Elad, M.; Sapiro, G. Sparse representation for color image restoration. *IEEE Trans. Image Process.* **2007**, *17*, 53–69. [[CrossRef](#)]
32. Ni, Z.; Ma, K.K.; Zeng, H.; Zhong, B. Color Image Demosaicing Using Progressive Collaborative Representation. *IEEE Trans. Image Process.* **2020**, *29*, 4952–4964. [[CrossRef](#)] [[PubMed](#)]
33. Glotzbach, J.W.; Schafer, R.W.; Illgner, K. A method of color filter array interpolation with alias cancellation properties. In Proceedings of the 2001 International Conference on Image Processing (Cat. No. 01CH37205), Thessaloniki, Greece, 7–10 October 2001; Volume 1, pp. 141–144.
34. Menon, D.; Calvagno, G. Color image demosaicking: An overview. *Signal Process. Image Commun.* **2011**, *26*, 518–533. [[CrossRef](#)]
35. Tan, R.; Zhang, K.; Zuo, W.; Zhang, L. Color image demosaicking via deep residual learning. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 793–798.
36. Cui, K.; Jin, Z.; Steinbach, E. Color image demosaicking using a 3-stage convolutional neural network structure. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 2177–2181.
37. Huang, T.; Wu, F.F.; Dong, W.; Shi, G.; Li, X. Lightweight Deep Residue Learning for Joint Color Image Demosaicking and Denoising. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 127–132.
38. Liu, L.; Jia, X.; Liu, J.; Tian, Q. Joint demosaicing and denoising with self guidance. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2240–2249.
39. Kokkinos, F.; Lefkimiatis, S. Deep image demosaicking using a cascade of convolutional residual denoising networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 303–319.
40. Kokkinos, F.; Lefkimiatis, S. Iterative Joint Image Demosaicking and Denoising using a Residual Denoising Network. *IEEE Trans. Image Process.* **2019**. [[CrossRef](#)]
41. Bean, J.J. Cyan-Magenta-Yellow-Blue Color Filter Array. U.S. Patent 6,628,331, 30 September 2003.
42. Lapray, P.J.; Wang, X.; Thomas, J.B.; Gouton, P. Multispectral filter arrays: Recent advances and practical implementation. *Sensors* **2014**, *14*, 21626–21659. [[CrossRef](#)]
43. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical image computing and computer-assisted intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
44. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
46. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
47. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
48. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]

-
49. Li, J.; Bai, C.; Lin, Z.; Yu, J. Optimized color filter arrays for sparse representation-based demosaicking. *IEEE Trans. Image Process.* **2017**, *26*, 2381–2393. [[CrossRef](#)] [[PubMed](#)]
 50. Chakrabarti, A.; Freeman, W.T.; Zickler, T. Rethinking color cameras. In Proceedings of the 2014 IEEE International Conference on Computational Photography (ICCP), Santa Clara, CA, USA, 2–4 May 2014; pp. 1–8.
 51. Condat, L.; Mosaddegh, S. Joint demosaicking and denoising by total variation minimization. In Proceedings of the 2012 19th IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012; pp. 2781–2784.
 52. Condat, L. A new color filter array with optimal properties for noiseless and noisy color image acquisition. *IEEE Trans. Image Process.* **2011**, *20*, 2200–2210. [[CrossRef](#)] [[PubMed](#)]
 53. Jeon, G.; Dubois, E. Demosaicking of noisy Bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *IEEE Trans. Image Process.* **2012**, *22*, 146–156. [[CrossRef](#)]
 54. Ye, W.; Ma, K.K. Color image demosaicing using iterative residual interpolation. *IEEE Trans. Image Process.* **2015**, *24*, 5879–5891. [[CrossRef](#)] [[PubMed](#)]